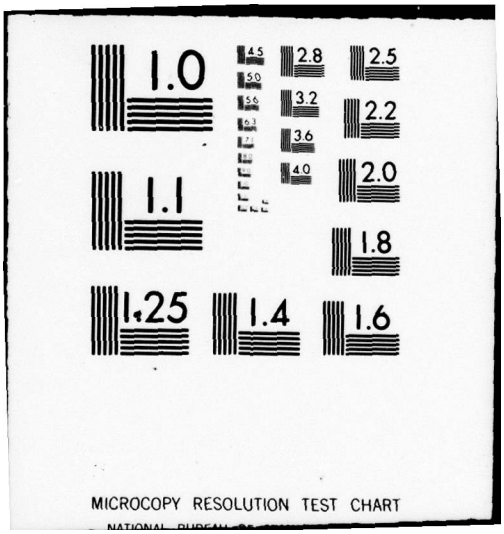
AD-A073 101 MITRE CORP BEDFORD MA F/G 9/2
USER REQUIREMENTS FOR COMPUTER SECURITY,(U)
MAY 79 J M SCHACHT, S M GOHEEN, R D RHODE F19628-78-C-0001
UNCLASSIFIED MTR-3596 ESD-TR-79-127 NL

1 OF 1
ADA 073101

END
DATE FILMED
9-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART

ESD-TR-79-127

MTR-3596

USER REQUIREMENTS FOR COMPUTER SECURITY

BY J.M. SCHACHT, S.M. GOHEEN, R.D. RHODE

MAY 1979

Prepared for

DEPUTY FOR TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 672B
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-78-C-0001

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Daniel R. Baker

DANIEL R. BAKER, Capt, USAF
Technology Applications Division

Charles J. Grewe, Jr.

CHARLES J. GREWE, JR., Lt Col, USAF
Chief, Technology Applications Division

FOR THE COMMANDER

Normand Michaud

NORMAND MICHAUD, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Technical Operations

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(19) REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) ESD-TR-79-127	2. GOVT ACCESSION NO. — 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) USER REQUIREMENTS FOR COMPUTER SECURITY	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER (14) MTR-3596
7. AUTHOR(s) (10) J.M. Schacht, S.M. Goheen, R.D. Rhode	8. CONTRACT OR GRANT NUMBER(s) (15) F19628-78-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P.O. Box 208 Bedford, MA 01730	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 672B
11. CONTROLLING OFFICE NAME AND ADDRESS (11) Deputy for Technical Operations Electronic Systems Division, AFSC Hanscom AFB, MA 01731	12. REPORT DATE (12) MAY 79 (54p.) 13. NUMBER OF PAGES 53
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

COMPUTER SECURITY
COMPUTER SECURITY REQUIREMENTS
MULTILEVEL SECURITY
SECURITY
SECURITY PLANNING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The various approaches to secure computer processing of classified information are summarized and contrasted. Dedicated processing, period processing, jobstream separation, multilevel security, and other approaches are characterized according to cost and risk factors, and data-sharing capabilities.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

235 050

JOB

ACKNOWLEDGMENT

This report has been prepared by The MITRE Corporation under Project No. 672B. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

BACKGROUND ON REQUIREMENTS

INTRODUCTION

The pressing need for information security and computerized systems has spurred numerous agencies, both military and industrial, to develop new techniques and/or approaches to attain computer security. The approaches currently used, based on physical security and isolation (or separation), create a significant overhead in terms of cost and computer availability. In particular, these approaches greatly impact military systems, where responsiveness to command and fulfillment of mission responsibilities relies on classified multilevel information.

Other, more advanced, approaches to security, that rely on internal access controls, will have a profound effect on future ADP security, as new technology and state-of-the-art improvements in computer security methods replace their antiquated predecessors. However, until such time as these new approaches become available, Air Force middle- and high-level managers, security officers, and system designers must be aware of the availability, use and cost tradeoffs associated with both presently-used and soon-to-be-available approaches to computer security. By accounting for the development of future approaches, one can incorporate the most cost-effective techniques available, without jeopardizing future system upgrade or reconfiguration to an alternative mode of processing.

Toward this end, this report serves as an introduction and recapitulation of various aspects of the computer security problem. Sections I and II provide background data underlying ongoing computer security efforts, and outline several basic requirements for computer security. Section III focuses on the development of secure multi-level computer operations and introduces the concept of the reference monitor, and the problems associated with certification. Section IV provides a detailed look at existing techniques, prototype models and conceptual approaches to computer security and attempts to assess the overhead cost of each technique. Section V describes how one can determine the existence of, and assess the complexity of, multilevel security requirements based on current or projected operational information flows (e.g., files and users) within an existing system.

BACKGROUND

In the past, and to a degree in the present, military and commercial computer systems have provided information security by employing techniques originally designed to provide physical security, namely protection from hazards such as fire, sabotage, and theft. As a result, computer sites were physically isolated from the outside. In addition, decentralization of information processing functions and dispersion of responsibility for the generation and protection of the information were common techniques employed to enhance physical security. Although these methods prevented buildup of centralized processing power and minimized the loss of information from various hazards, it was found that physical security alone could not provide total information security. Protection from outside threats does not preclude security compromise from within the system. Since it was possible to control access to the information content stored in the system by physically controlling access to the storage medium itself, access privileges were granted to the storage medium, not to the information contained therein.

Since the inception of computer processing systems, computer services have been geared to providing access to information, rather than restricting it from particular segments of the user community. Also, as great strides were made in the development of computer technology, computer systems were relied on more heavily and were integrated into more complicated, sophisticated systems that handled sensitive or classified data. Unfortunately, the development of adequate computer security techniques has not kept pace with the development of advanced technology. Had serious attention been placed on computer security in the past, during the development of earlier computer system architectures and operating systems, the "problem" of computer security would, at this time, be less complicated and costly than it is. At present, military, governmental, commercial and industrial user organizations are faced with the burdensome task of developing approaches to computer security that will either be added on to (or retrofitted into) existing third generation systems, or be incorporated into the design of hardware and software components of future systems.

Third generation computers have introduced new capabilities that involve the concurrent processing of many jobs, extensive sharing of computer resources, and the use of remote terminals. While these new capabilities brought benefits of substantially lower cost, sharing of large data bases, and remote use of computers, they also increased the complexity of the security problem. The possibility of inadvertent, accidental, or malicious acquisition of information has increased significantly along with the use of multiprogramming/multiprocessing computer systems. Expanded system

capabilities accompanied by increased demand from a greater volume of users has compounded the security problem and called attention to the pressing need for the provision of adequate controls. Although some safeguards have been implemented, computer security in its current form has been shown to be inadequate by the ease with which system penetration tests have succeeded in capturing what was thought to be protected information. Thus, a malicious user may employ sophisticated techniques for penetrating the defense of a system by bypassing or suspending security controls, denying use of the system to others, gaining unauthorized access to information in it, and falsifying or destroying information.

In order to satisfy continually increasing data processing requirements while keeping costs in line, it has become imperative to institute more flexible modes of computer operation. These new modes include the running of programs at several different classification levels concurrently on the same machine, the providing of on-line interactive service to users cleared to a variety of security levels (possibly including uncleared users who may pose security threats to the system), and the establishing of communications links between different military computer systems to form computer networks. As each of these improvements in computer utilization efficiency is approached, a new problem will repeatedly arise: it will no longer be possible for the installation personnel to maintain data security without the active assistance of the computer system itself.

At present, several efforts directed at providing logical security enforcement in computers are underway or have been proposed. As a means of defense against penetrations, in addition to providing elements of protection resulting from a combination of personnel, physical, administrative, communications, and emanation security, various methods for processing classified material have been implemented. Section IV focuses on the known approaches to meeting computer security requirements. These approaches are categorized as to whether their protection features are internal or external to the system itself.

SECTION II

REQUIREMENTS FOR SECURE OPERATION

INTRODUCTION

This section details, on a high level, the issues of computer security and the overall design requirements for ADP systems in regard to security. In addition, an attempt is made in the latter part of the section to indicate the relative impact of various operational characteristics on security requirements.

THREATS

Threats against computing systems can be classified into three categories:

1. denial of service to others,
2. unauthorized disclosure (acquisition) of data, and
3. unauthorized modification of data.

Computer security is achieved when sensitive information is safeguarded from all threats. The categorization above assumes that the perpetrator is a determined individual who is sufficiently resourceful and knowledgeable to utilize one of the forms of threat. If he were not, then the problem of providing computer security would be reduced to preventing "accidental" penetration.

Protecting computer systems from the threats and vulnerabilities identified earlier depends largely on the degree of complexity of those systems; for example:

- The security problems are simplest when the computer systems are contained in secure areas.
- Computer systems that offer fewer services, particularly in a terminal environment, have simpler security problems.
- Use of remote terminals complicates the security problem; for example, they involve indirect identification rather than face-to-face recognition.

Denial of Service

By implementing one of many techniques from this threat category, the malicious penetrator inhibits normal system operation, thereby depriving the general user population of computer service. The intent of the penetrator is not to steal or modify specific information, but to stop the authorized user from accessing his own data, by introducing annoying delays, bottlenecks, and service interruptions. Service interruption techniques include crashing the operating system, or generating numerous time-consuming requests to degrade system response.

Unauthorized Disclosure of Information

This threat category involves gaining unauthorized access to read sensitive information. The copying or stealing of such information is commonly thought of as the computer security problem. The section entitled 'Vulnerability' will focus on the techniques used to gain access.

Unauthorized Modification of Information

This last threat category is probably the most insidious of all. By manipulating or modifying the contents of a data base (e.g., gaining write-access to disk), the penetrator can create a situation where erroneous data can unknowingly be used in sensitive computations to create devastating consequences.

FACTORS AFFECTING COMPUTER SYSTEM SECURITY

This subsection presents an approach to comparing alternative computer configurations according to the magnitude of the security problems they present. The approach consists of describing each configuration in terms of a set of attributes. The relative ease of providing security for any pair of configurations may be determined by comparing the values of the attributes that apply to them. The attributes do not provide a method of assessing quantitatively the security risk presented by a given configuration. Furthermore, the attributes resemble dimensions in some multi-dimensional space; thus, there is no direct way of comparing configurations that differ in two or more attributes. However, the use of the approach outlined below should at least facilitate the orderly comparison and evaluation of alternative configurations from a security viewpoint. The following describes several attributes that are used to categorize configuration security and presents the range of values of each.

SYSTEM SECURITY ATTRIBUTES

The eight attributes that are used to characterize computer system security requirements are:

1. maximum data (or program) classification level,
2. minimum user area (or user) clearance level,
3. level of user control over system operation,
4. complexity of system mode of operation,
5. requirement for need-to-know control and user identification,
6. integrated (multilevel) processing,
7. external technical characteristics, and
8. internal technical characteristics.

The following paragraphs describe each of the attributes and its alternative values. Table I summarizes the attributes and their values.[1]

Data Classification Level

The first attribute used to characterize computer configuration security is the maximum classification level of data or programs present within the system. This attribute was considered important on the assumption that, all things being equal, one would rather use a given system for processing data of a lower classification than a higher one. For example, a certifying official might consider a

---

[1] The regulation governing the classification, downgrading, declassification and safeguarding of classified information is DoD ISPR (Information Security Program Regulation) 5200.1-R. The regulation amplifying those policies for use within the Air Force and providing procedural details where appropriate is AFR 205-1. Together, these regulations form the foundation for all Air Force policies and procedures regarding classified information. The rules and regulations pertaining to classified computer processing are described in DoD manual 5200.28-M "Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating, Secure Resource Sharing ADP Systems".

time-sharing system acceptable for secret data, but not for top secret. The "multilevel security" problem presented by a system is reflected by a combination of high data classification and lower user clearance level, as discussed below.

The values taken on by the data classification attribute are the four usual classification levels (unclassified through top secret) plus "top secret codeword". The latter value may be used to reflect the presence of data such as SIOP requiring special handling and precautions.

User Clearance Level

The minimum security clearance level of any user or user area supported by a computer system is a major factor in determining the risk of compromise that the system presents. A system supporting uncleared users at unsecured terminals presents a maximum risk of compromising any stored classified information. In contrast, a system whose users are all cleared at or above the secret level seems to present a relatively low risk of serious compromise; presumably a cleared user is fairly likely to report any inadvertent disclosure of top secret data.

The magnitude of the security problem that a system presents is dependent on both data classification and user clearance levels. A system with unclassified data and uncleared users presents little or no problem; one with some top secret data and uncleared users presents a serious problem. In comparing alternative configurations, an analyst must consider both attributes together and develop a "classification-clearance range" attribute. This range may indicate problems in a nonlinear manner. For example, either the presence of uncleared users or that of top secret codeword data complicates matters to a considerable extent.

The range of values for user clearance level is the same as that for data classification level. The approach outlined above assumes that a user, his terminal area, and the communications to the terminal are all cleared to the same level. If that assumption does not hold, the lowest of the three governs the risk of compromise.

Level of User Control

The level of user control over a computer system is an indication of the likelihood that a user program will, deliberately or inadvertently, capture the system and compromise classified information. If a system simply accepts data and processes it, the system's software must only be certified to cause no compromise itself. To

TABLE I

Security Attributes

INSTALLATION REQUIREMENTS

LEVEL OF USER CONTROL

1. Minimum Control (Data Input Only)
2. Storage/Retrieval Language
3. Interpretive Programming Language
4. Compiler Programming Language

COMPLEXITY OF SYSTEM OPERATION

1. No Multiprogramming or Data Base
2. Data Base But No Multiprogramming
3. Single Application Multiprogramming
4. General Multiprogramming

SECURITY REQUIREMENTS

MAXIMUM DATA CLASSIFICATION LEVEL

1. Unclassified
2. Confidential
3. Secret
4. Top Secret
5. Top Secret/Categories

MINIMUM USER CLEARANCE LEVEL

1. Unclassified
2. Confidential
3. Secret
4. Top Secret
5. Top Secret/Categories

NEED-TO-KNOW CONTROL AND USER IDENTIFICATION

1. No Need-To-Know Control
2. Need-To-Know by File and Terminal
3. Control by Individual User

TABLE I (Concluded)

INTEGRATED (MULTILEVEL) PROCESSING

1. Isolation
2. Multilevel Sharing of Information (minimum user clearance)
3. Multilevel Sharing of Information (open operation)

TECHNICAL REQUIREMENTS

EXTERNAL

1. Dedicated Systems
2. Periods Processing
3. System High
4. Controlled Environment

INTERNAL

1. Jobstream Separator
2. Virtual Machine Monitor
3. Multilevel Secure System

the extent that user programs are present, or that user inputs control system action, the software must also be certified capable of protecting itself against the user.

The values of the user control attribute range from simple data input through query languages to several types of programming is shown in Table I. The values are intended to show the range of user control and to reflect the probable difficulty of certifying system software. The distinction between assembly language and compiler language programming may not be significant in some systems. In most current computer systems, however, the assembly language programmer has somewhat more control than the programmer who works in a high-level language.

Complexity of Operation

The attribute, "complexity of system operation", is intended to show the extent to which the system must simultaneously manage multiple levels of classified information. In particular, complexity is indicated by the extent of multiprogramming in the system and by responsibility for a permanent (on-line) data base. A system that processes only one job at a time with all files changed before and after processing clearly has little inherent risk. One that manages a data base for many users and multiprograms several applications together faces a considerable challenge. Between these extremes are the cases of systems with on-line files but no multiprogramming and with multiprogramming of transactions against a single application package. A batch processor with on-line disk storage would fall into the former category and an airline reservation of communications processing computer into the latter.

Need-To-Know Control

Need-to-know control and user identification complicate the task of security control by creating classes of data subordinate to the usual clearance levels and requiring additional checks on data access. The least complicated system is one with no need-to-know controls. Next comes a system where terminals are assigned to classes with differing access rights. Most complicated is the situation where individual users must be identified and their access rights remembered and enforced.

Integrated (Multilevel) Processing

This topic is discussed extensively in Section III.

Technical Characteristics

Section IV discusses internal and external technical requirements and approaches.

SECTION III

MULTILEVEL COMPUTER OPERATIONS

INTRODUCTION

This section serves as an introduction to the only demonstrated technical solution to the problem of multilevel computer security. This approach is based on the reference monitor/security kernel concepts. A brief review of the development history is provided, as is a technical description of the approach. In addition, the issues of technical and adminstrative certification of the system for multilevel operation are discussed.

BACKGROUND

The notion of multiple levels of sensitivity and protection is common to most systems which seek to restrict the dissemination of information. Multiple levels arise because it would be overwhelmingly expensive to protect to the maximal extent all material requiring any protection, and it would be decidedly impractical to clear to the highest level everyone who requires access to any protected material. Therefore, a set of levels is defined that permits a practical degree of protection to be provided for information, corresponding to its degree of sensitivity.

The United States military security system defines levels of sensitivity by using two variables. The first is classification, a hierarchical set including (but, under special circumstances, not limited to) four levels: Unclassified (i.e., requiring no protection), Confidential, Secret, and Top Secret (i.e., requiring maximal protection). By hierarchical, it is meant that the four classifications are ordered, and clearance to one level implies clearance to all lower levels. The second variable is access category, a non-hierarchical set orthogonal to classification. Access categories are not fixed in number; new ones may be created and old ones may be terminated. At present, they include material protected at the request of foreign powers (e.g., pact organizations such as NATO and SEATO, as well as the governments of individual foreign nations), material protected under the authority of non-military agencies (e.g., Restricted Data under the Atomic Energy Commission, CRYPTO material under the National Security Agency, etc.), and material associated with specific restricted access activities entirely within the military.

Standards of physical protection are defined for material at each level of sensitivity. For items at specific classifications and in no special access categories, the levels of physical protection correspond exactly to the classification levels. For an item at a particular classification and in a particular category, the level of physical protection mandated is typically somewhat more elaborate than that for no-category items of the same classification, but still not adequate for the protection of material at the next higher classification. Thus, the United States military security system can be spoken of as assigning every item to one of a single ordered set of physical protection levels, with those levels defined by the two security variables, classification and category.

Of course, it is not a security violation if material at a particular sensitivity level is physically protected to a level higher than the mandated one. However, making this sort of over-protection a regular practice is deemed undesirable, primarily for two reasons. First of all, higher physical protection levels are always more expensive to create and maintain. Therefore, the over-protection of significant amounts of material generally represents a serious waste of limited funds and resources. Secondly, continual over-protection can often lead to carelessness on the part of personnel responsible for maintaining security.

Computer Systems

These considerations have a direct bearing on the operation of data processing installations that handle classified information. The physical protection accorded the computer itself must, of course, correspond to the highest sensitivity level of information that can ever be processed by the installation. Some input/output devices, however, may be restricted to processing information only at levels well below the installation's overall clearance, and a correspondingly lower level of physical protection should be provided for these devices. Indeed, the protection provided for each individual item of I/O material should only be as stringent as is dictated by the sensitivity of the item's contained information. This philosophy implies that the installation personnel should always be cognizant of the level of physical protection which is appropriate to each item of I/O material which they handle.

Contemporary computer systems in general do not include logical security enforcement mechanisms. As a result, they must process information at only one sensitivity level at a time, in order to avert the possibility of unauthorized persons obtaining classified information by essentially instructing the computer to give it to them. Systems which operate under these conditions may be called

unilevel systems. One characteristic of a unilevel system is that it must undergo sanitization when the sensitivity level of its computational load changes. In this environment, it is an easy matter for installation personnel to know the sensitivity level of all I/O material, since each I/O device can at any instant only be processing material at the level of the system itself. If proper procedures have been followed, all other material has been removed from the machine room or locked in appropriate storage.

Newer systems will incorporate logical security enforcement mechanisms. The actual amount of increased operating flexibility obtained will depend on the comprehensiveness and reliability of the mechanism employed. In any case, such systems will be multilevel systems, authorized to concurrently perform computation at more than one sensitivity level.

A UNIFIED TECHNICAL APPROACH TO MULTILEVEL COMPUTER SECURITY

This subsection introduces the foundation of the computer security development effort. It describes the history and origin of the technical approach; briefly summarizes the approach and its main implications; and discusses the technique for verifying the security of a computer system that solves the problem of completeness.

The Computer Security Technology Panel

In 1970, the Air Force Data Services Center (AFDSC) asked the Electronic Systems Division to support development of open multilevel secure operation for AFDSC's Honeywell 635 computer systems. The 635's operate under control of the standard GCOS III operating system. ESD and MITRE personnel shortly reached conclusions substantially identical to those given above: that no set of modifications to GCOS III would render it suitable for multilevel operation, much less for open operation with uncleared users and terminals.

To determine the reasons for the difficulty with GCOS III, and to identify ways of solving future multilevel security problems, the Air Staff directed ESD to convene a computer security technology planning study panel. The panel, composed of recognized experts from industry, universities, and government organizations, convened in early 1972. The panel operated under a contract from ESD to James P. Anderson and Company, and was tasked with preparing a development plan for a coherent approach to attacking the problems of multilevel computer security. The panel was supported by a working group of computer system staff officers from ten Air Force commands who identified the operational and economic impacts resulting from the lack of computer security technology.

The panel identified the problem of completeness and recognized the futility of "patching holes" in existing operating systems. It recommended as a technical approach "to start with a statement of an ideal system, a model, and to refine and move the statement through various levels of design into the mechanisms that implement the model system".[2]

THE REFERENCE MONITOR

The basic component of the ideal system proposed by the security technology panel is the reference monitor--an abstract mechanism that controls access of subjects (active system elements) to objects (units of information) within the computer system. Figure 1 schematically diagrams the relationships among the subjects, objects, reference monitor, and reference monitor authorization data base. The figure gives examples of typical subjects, objects, and data base items.

An implementation of the reference monitor abstraction permits or prevents access by subjects to objects, making its decisions on the basis of subject identity, object identity, and security parameters of the subject and object. The implementation both mechanizes the access rules of the military security system and assures that they are enforced within the computer.

The security technology panel stated that the reference monitor must meet the following three requirements:

1. Completeness - The implementation must be invoked on every access by a subject to an object.

2. Isolation - The implementation and its data base must be protected from unauthorized alteration.

3. Certifiability - The implementation must be small, simple, and understandable so that it can be verified to perform properly.

The requirement for completeness demands that the implementation of the reference monitor include both hardware and software to avoid the complexity and overhead that would result from software validation of every access. The requirement for certifiability makes the same

---

[2] J.P. Anderson, Computer Security Technology Planning Study," ESD-TR-73-51, Vols. I and II, Electronic Systems Division, AFSC, Hanscom AFB, MA, October 1972 (AD758206 and AD772806).

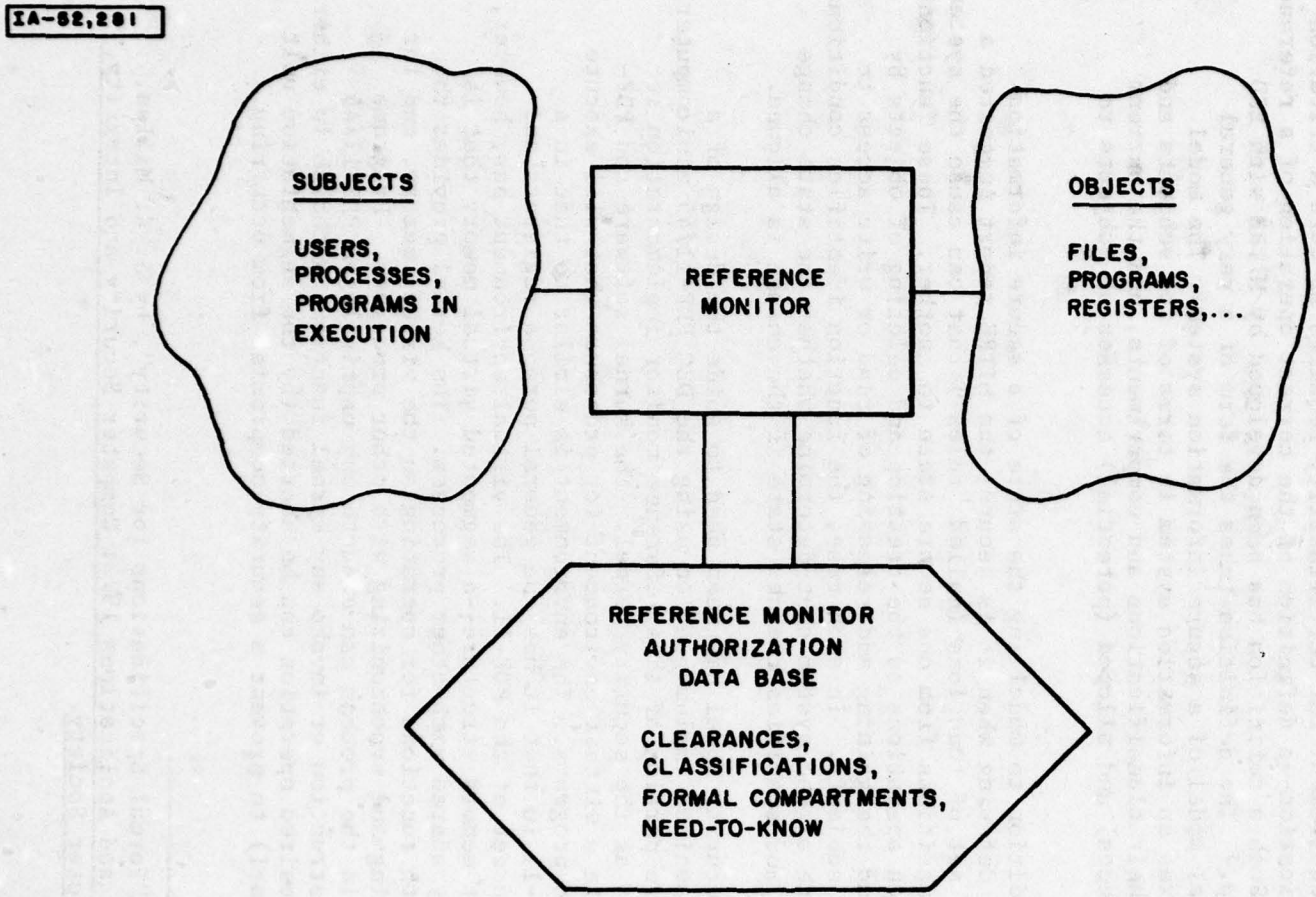


Figure 1. THE REFERENCE MONITOR

demand because certain hardware architectures preclude the implementation of a small, simple reference monitor.

The requirement for certifiability requires in turn a criterion for certification--a definition of the correct operation of a reference monitor. Such a definition has been developed by MITRE with ESD sponsorship.[3] The definition takes the form of a very general mathematical model of a secure information system. The model characterizes an information system in terms of its subjects and objects, their classifications and compartments, and the current (instantaneous) and allowed (potential) accesses of subjects to objects.

In addition to modelling the state of a secure information system and defining when it is secure, the MITRE report suggested a particular set of functions (called "rules") that can cause the system to make transitions from one secure state to another. These functions express such operations as the creating and deleting of objects by subjects and the gaining and releasing of read or write access to objects by subjects. In each case, the function identifies conditions on the state of the system that determine whether the state change is allowed and specifies the new state if the change is allowed.

The security model has been used to guide the design of a reference monitor implementation using the DEC PDP-11/45 minicomputer. The software portion of this reference monitor implementation is referred to as the security kernel. The kernel software and PDP-11/45 create a virtual environment for processes that will execute uncertified programs. The environment is similar to that in a "bare" PDP-11 in that it has the general purpose registers and instruction set of the PDP-11. The virtual environment has, however, a different memory structure--a segmented virtual memory that is selectively shared with other processes. The kernel provides the process with functions for operating on the virtual memory, and for communicating and synchronizing with other processes. Programs executing in the process can execute any unprivileged PDP-11/45 machine instruction or invoke any kernel function, although in either case the desired operation can be aborted (by the segmentation unit or the kernel) to prevent a security compromise from occurring.

---

[3] See also "Formal Specifications for Security", by J. K. Millen, in Trends and Applications 1977: Computer Security and Integrity, IEEE Computer Society.

The reference monitor design implements the subjects of the model as processes and the objects as segments, input/output devices, and interprocess communication channels. The kernel program includes an entry point corresponding to each function suggested for the model. In addition, two other classes of functions are required fit the reference monitor implementation into an environment of finite hardware with limited functions. The first added class of function provides for altering the representation of the current security state--for example, by multiplexing one processor among many processes. The second added class provides interpretive access to those objects like interprocess communication channels whose direct reading and writing cannot be adequately restricted by the PDP-11/45 hardware.

The technical verification that an implemented reference monitor is a valid interpretation of the security model requires a formal proof. The proof approach that has been developed[3] requires the description of the reference monitor by a formal specification. Such a specification can be proven to satisfy the security properties of the model. The specification, in turn, imposes requirements for the proof or testing of the reference monitor software and hardware.

Models and Technical Validation

Recognizing the importance of the panel's "ideal model" as a starting point, ESD initiated development of a mathematical model of computer security in 1972. Preliminary efforts were performed in-house and subsequent contributions were made by The MITRE Corporation and by Case Western Reserve University.

The completed model of secure computer systems represents a secure computer system as a finite-state mechanism that makes explicit transitions from one secure state to the next. The state of the system is defined by:

a. the classifications and compartments of all subjects and objects;

b. the need-to-know relationships of subjects and objects;

c. the hierarchical organization of objects (in a storage system); and

d. subjects' current ability to access objects.

The rules suggested for the model formally define the conditions under which a transition from state to state can occur. The rules are proven to allow only transitions that preserve the security of information in the system.

A significant property of the model is that all but a special collection of "trusted" subjects are restricted from writing information at a lower classification (or proper subset of compartments) than they read. The restriction prevents information obtained at the higher level from being transferred to a lower level where it can be accessed illegally. This property eliminates the need to verify that all programs (such as editors and utility routines) do not act as "Trojan Horses"[4] and downgrade classified information.

The model of secure computer systems specifies requirements for the operation of a security kernel. The requirements identified by the model are taken directly from the Defense Department regulations on handling sensitive information (DoD Directive 5200.1-R). The problem of validation is then reduced to providing complete assurance that the security kernel behaves as the model requires.

For some time after the basic security model was developed, there was doubt as to the appropriate technical approach to providing the assurance mentioned above. In 1973 it was recognized that the work of Price[5] identifies a methodology for providing the required assurance. This methodology involves preparing a formal specification for each function of the security kernel. The collection of specifications is then proven to be internally consistent and to satisfy the security properties of the model. The descriptions of the functions in the specification language are close to a programming language and facilitate proof or verification of the code that implements the specified kernel design.

While the basic methodology developed by Price applies to validation of small security kernels (up to perhaps 1000-line computer programs), the implementation proof may become cumbersome for larger kernels. Therefore, a hierarchical specification and proof technique

---

[4] A Trojan Horse is an apparently innocent but actually malicious computer program that is typically developed by one individual for use by another. When the program is operating on behalf of the intended user, it accesses that user's sensitive data, then makes it available to the program's author (for example, by writing it in a "hidden" file).

[5] W. R. Price, "Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, June 1973.

that divides the specification modules into manageable subsets is being explored in addition to the basic Price methodology.

The paragraphs above have summarized the basic elements of ESD's approach to the design and technical validation of secure computer systems and security kernels. While the administrative certification that a computer is secure must be based on formal policy, it is likely that a technical validation approach such as that outlined provides the only adequate basis for such formal certification.

CERTIFICATION

People understand the need for security and the consequences, both to the nation and to themselves, of compromising security; machines do not, and cannot be expected to in the future. A major problem associated with the development of secure computer systems is the controversial issue of certification, which may be defined as "the confirmation that protection capabilities of a computer system are compliant with technical requirements for security". It will be possible to decide whether or not security, as defined by requirements, is attained, only if there is an effective methodology for determining compliance of the system with the requirements for a secure computer system.

The technology to produce provably correct computer programs exists, and the methodology for applying this technology to the creation of certifiable logical security enforcement mechanisms is being developed. These advances will permit appropriate officials to guarantee, on their personal authority, that various security-related functions will be performed correctly by a computer system. In this sense, software can be certified, and certified software may be said to be "responsible" for security enforcement, just as an approved safe may be said to be "responsible" for the physical protection of classified documents. Certification raises technological and administrative problem areas. The most significant and pervasive of the problems recognized to date are:

1. the difficulty of the certification process, and

2. the impact of system complexity on certification.

Certification and Testing

If an organization is to operate a computer system with multiple levels or types of classified data and with a community of users having diverse clearances, it must undertake to certify that the

system restricts each user to that data which he has need to know. Evolving regulations and directives provide an organizational framework for certification, but little technical guidance that would help determine a system's adequacy. Some certification attempts now under way are based on the concept of a test team or penetration team.[6] Such a team is given access to the system and its documentation and directed to "find the holes". Thanks to the underlying weakness of most current systems, test teams have been quite successful in finding "holes". In such cases, certification is usually denied and the system continues to run in a closed (not multilevel) environment.

The danger and difficulty of the penetration team approach comes from the question, "how much is enough?". If a test team finds one or five or a hundred ways of penetrating a computer system, one can say with certainty that the system is not secure. But if the team finds no holes, or if the system builders repair the ones discovered, that does not imply that the system is secure. The strongest statement that can then be made is that the holes (if any) in the system are well hidden. The operator of such a system is betting that either:

1. there are no holes in his system; or

2. a potential attacker is less clever and persistent than the test team.

With current complex systems (see below) the first possibility is an unlikely one.

If a system is certified "hole-free" by an ad hoc test approach, its operator must ask about the impact of changes and updates. If one has certified a computer system (hardware and software) as an integrated whole, any change requires recertification of the entire system (as in AUTODIN). While a very simple change may have a restricted and obvious effect, changes of larger scale rapidly develop the ability to obscure their effects--both functional and security effects. Thus, at some ill-defined point, one must start certification from the beginning after a change has been implemented.

The discussion so far points out the difficulties of certifying the security of any system, and the special problems of recertifying a highly integrated system. Clearly, one would like an orderly

---

[6] Sometimes called a "tiger team".

approach to certifying a system so that it could be confidently asserted that the certification was complete. An approach to isolating security controls from functional parts of the system would help by allowing functional changes and evolution to proceed without requiring recertification. If security controls could be minimized as well as isolated, certification or testing would also be simplified since the security controls would tend to become obvious and understandable. However, the isolation must be absolutely complete and effective, lest one leave in the security controls a trap door suitable for entry from any point in a large uncertified portion of the system.

Complexity

The second major problem associated with the development of secure computer systems is the influence of complexity in the development of uncertifiable computer systems. In this context, complexity refers to the tendency of large interrelated blocks of software to become inseparable from a system's security controls. Thus the would-be certifier must understand every state and function of programs that (he hopes) have nothing to do with security. The danger is that one such program will in fact have a (negative) security function.

Complexity in modern computer systems results mainly from the hardware on which those systems are built: while one might build inadequate software on better hardware, the bulk of existing hardware simply fails to support anything but complexity. Typical third-generation computer systems provide some form of memory protection (write and read) and relocation, plus a set of two processor states (privileged and user). In the more privileged processor state, a process (program in execution) can issue input/output instructions, reset the memory protection, and control the entire state of the processor. In user state, a process can only execute ordinary arithmetic and logical instructions on data within its memory partition. Some low-cost minicomputers do not even provide the level of protection described above.

The provision of only two processor states in conventional computers makes more difficult the isolation and minimization of security controls. While operating system designers might put operating system functions requiring protection but not privilege in separate bounds-protected programs, current hardware does not provide convenient communication between a user program and such a separate operating system routine. Such communication must be mediated (at some cost) by the privileged portion of the operating

system. The cost and difficulty of this approach are so great that most operating system functions wind up in the privileged portion of the executive.

Finally, the restriction on input/output instructions requires that an elaborate system of input/output control programs be included in the privileged portion of the operating system. These programs are complex, and have the upsetting habit of requiring modifications whenever a new input/output device is to be added to the computer.

The preceding paragraphs have illustrated the way in which inadequate hardware leads to overly complex operating system software. Practical illustrations of this effect are provided by most current commercially available operating systems.

Off-the-Shelf Software

The discussion above focused on the complexity that appears in an off-the-shelf computer's operating system as a result of the computer's architecture. To this complexity is added a basic lack of consideration for security in operating system design. The result of this lack is often increased complexity coupled with dispersion of security controls throughout the operating system.

A typical modern operating system is based on a set of interconnected tables or list structures. These list structures describe the state of tasks in execution, main memory, input/output devices, and files on secondary storage. Any change in the status of a process is reflected in one or more tables. The security problems associated with such a structure are twofold:

1. Almost every table and table entry has some implicit or explicit security role.

2. Tables are accessed "as needed" from all parts of the operating system.

In addition to these two underlying problems, modern operating systems typically have no consideration of such notions as classification, clearance, or need to know. The two underlying problems result in potential security violations that are legal system operations. (For example, read a top secret file and write its contents to an unclassified file.)

Modifying an off-the-shelf operating system for security is a grim and ineffective business. At best it requires an examination and rewrite of all security related code--usually the entire operating

system. Such efforts have been marked more by their cost than their security. At worst a modification may put a thin veneer of "security features" over a deep, complex, and unsecure structure. Such features may be suitable for labeling output, but typically provide no protection.

A final software aspect of computer security concerns languages for system implementation. Most operating systems and communications programs have historically been written in assembly language. In addition to its known disadvantages for training, documentation and maintenance, assembly language provides a ready vehicle for obscuring the security implications of a program. The alternative, higher-level language for system programming, is often claimed to be costly of space and time. However, most recent experience and evolving commercial practice tend to contradict these claims.

SECTION IV

APPROACHES

A number of methods for secure processing of classified information currently exist or are being planned for use in Air Force environments. These methods are based on AFR 300-8 which identifies three basic protection modes:

1. dedicated mode,
2. controlled environment, and
3. multilevel processing mode.

In the following presentation, another dimension--focusing on the external/internal location of the protection features of the system--will be used to describe the basic approaches to secure processing design.

Basically the security functions that must be guaranteed by any secure operating environment relate to two key operations:

- segregation of different levels of classified data, and
- authorized control over user's access to appropriate levels of data.

The external/internal variable refers to the implementation of these control functions through external physical and administrative control procedures, or alternatively via automated (hardware-software) mechanisms internal to the computer system.

Among the external approaches identified here are:

- dedicated systems,
- periods processing,
- system high, and
- controlled environment.

The internal built-in protection approaches include:

- job stream separator
- virtual machine monitor
- multilevel secure operating system

In addition, it is suggested that, in fact, user requirements may not match any one of these approaches entirely and some combination of the above modes may be the best solution. Finally, the security problem may appear in a network of computers. Some of the main factors relating to network security are also discussed.

The following section gives a brief description of all of these approaches to computer security with emphasis on the economic considerations affecting the user of each mode of secure operation. The cost factors and formulas presented here are for basic planning purposes only, but should give a broad basis for analyzing different approaches to the problem. In addition to the cost factors discussed, the user should consider relative risks (security weaknesses) alluded to in the description of each approach. Potential weaknesses in the approaches indicate special attention may be given to these areas.

EXTERNAL APPROACHES

All external approaches place heavy demands on setting up various kinds of safe external environments. The dedicated mode (which includes dedicated system, periods processing, and system high) basically assures that all users of a system have clearances equal to or greater than the highest classification of information to be processed. The separation of users to one level is satisfied by controlling one of three variables: separate machines, separate time periods, or full clearances to all users of the system. Controlled environment, on the other hand provides a limi ed multi-level capability while still relying on the external environment to maintain the required degree of security. The latter is provided through several means all centered around some degree of limiting the multilevel capability--as in application of uni-directional communication interfaces to provide limited integrated processing with data passing only from the lower levels to a high level; or enhancing operating system controls to support a "controlled" (two-level) sharing that excludes unclassified users.

Dedicated Systems

A dedicated system approach to processing multiple classifications of data introduces physical isolation to separate the user and data on one level of clearance from all other levels. Each secure

level of operations has a full set of hardware such that all data are at a single classification level and all users of that system have clearances greater than or equal to the highest classification. For a multilevel information system, the dedicated systems approach simply redefines the problem to create multiple single level environments. This technique provides reliable isolation of data classification levels and outside restrictions on user access at the expense of:

a. supporting multiple machine facilities, and

b. inter-level data sharing.

Cost Factors

From an economic point of view, the expense of operating machines dedicated to individual levels of classification can be measured in terms of extra machine capacity. Though one might typically try to match system capacities to the workload at each level, the maximum loss of dedicating two systems where only one is required to do the total job operating in a multilevel mode is the whole cost of the second system.

Periods Processing

For periods processing, the isolation of various levels of classified data is based on separate time intervals dedicated to a single classification and clearance level. Only one level may be active at a time and each level constitutes a "period". The requirement for data sharing must be met by authorizing duplicate copies of classified data to be run during another level's time period. The requirement of purchasing and maintaining dual sets of hardware, encountered in dedicated systems approach, is replaced by a scheduling requirement that may reduce the responsiveness of the system to a particular level's turnaround needs. Response time requirements are further infringed by the "color change" time required to transfer operations mode from one level to the next. Each time change to a new level occurs, time must be deducted from actual production processing in order to dismount media and clear the system of any program and data traces from the previous level. Manual procedures are used to effect the transition between processing periods (i.e., security levels)--clearing processors and memory of sensitive data, removing or replacing demountable storage media, physically clearing the facility of all sensitive material, and rebooting the system with a new copy of the operating system.

Cost Factors

The security-related expense for operating in a periods processing environment is measured in terms of time lost during period changes. During any one change period there is considerably more efficiency lost than is apparent in the actual shutdown time. Depending on how each installation handles it, such things as system status parameters may have to be saved or lost, thus affecting programmer efficiency on each system restart.

Based on an average of two transitions per day and a minimum 30 minute change period, the cost of periods processing is at least 4.1% of system cost. Various estimates put the actual figure in a range between 4 and 25%.

System High Operation

Procedures for system high operation dictate that all users be cleared to the highest level and treat all information being processed by the computer as though it had the highest classification. Under system high operations, all levels of data become temporarily classified at one level, thus enabling all programs and data to concurrently use the same hardware resource and engage in unlimited exchanges of data. Data reclassification decisions must be made during a downgrading process following each computer run.[7] Theoretically, errors in this process would not result in a high security risk as the entire user community is cleared to the highest level. Drawbacks of this approach are the cost of issuing clearances to a large number of users and the vulnerability of the system to careless programming and to Trojan horses whose intent is to maliciously modify classified data files.

Cost Factors

The major security expense for a system high operation is the cost of processing clearances beyond necessary levels. This includes not only initial extra clearances, but also calculations for personnel turnover. In addition, the real cost of obtaining excess clearances may not be an economic factor to begin with; there is always an increasing risk of compromise as the circle of cleared users expands.

---

[7] In the context of system high operation, downgrading is a process whereby data assigned a higher level for computer processing is reassigned its original security level.

Finally, in creating a processing environment that operates at the highest level at all times, all remote sites and communication links now require maximum area protection whether or not they process information classified at the system high level.

The actual number of new clearances needed to prepare any given system for a "system high" operation are specific to a particular user environment. Samples indicate the number is anywhere from 96%[8] to 20% of users, excluding programming and computer support personnel. The average cost of obtaining higher clearances is difficult to establish, but may be anywhere from $100-300 for SECRET and $300-1000 for higher clearances.

Controlled Environment

There are a number of ways to create a controlled or safe environment for running classified operations. While the system high approach creates a safe environment on one level, the controlled environment tries to achieve a safe multilevel environment. The difference between this and true multilevel approaches is the degree to which reliance on internal computer security controls occurs. Even with a system enhancement that significantly increases the reliability of security controls within the operating system itself, the internal mechanisms may still not be certifiably correct. Therefore, as with all cases of controlled environment, some aspect of the user's external environment must restrict and limit (e.g., constrained to two level, requiring a minimum clearance level) full multilevel use in order to assure adequate protection. Some of the other ways safe or controlled environments can be created include:

- Uni-Directional Communication Interfaces - which only allow uni-directional flow of information to a receiver equal or of higher clearances than the sender.

- Restricted Query/DMS Language - which defines a limited set (Query or DMS language) of user commands and dedicates all multilevel processing time to the query/DMS language operations (viz, restricts user programming or program development applications).

[8]Where as much as 96% new clearances are required, classified processing might better be handled by some other approach.

Cost Factors

Again, for a controlled environment the costs of external protection measures are reflected in excess clearances and remote site security controls.

INTERNAL APPROACHES

All internal approaches, (which include in addition to multilevel integrated processing the jobstream separator and virtual machine monitor designs) are designed to reduce the frequency and significance of error-prone human intervention. The internal designs are based on a new technology with the primary protection mechanism located internally in the system hardware and software components.

Briefly, the internal technology relies on control functions implemented through hardware and software components of the system. The technology centers around the concept of a reference monitor, a mechanism that:

- mediates all access attempts,
- is protected from the remainder of the system, and
- is provably correct.

The software portion of the reference monitor is the security kernel. The operation of the reference monitor is described by a mathematical model that is based on DoD regulations for handling sensitive information (DoD Directive 5200.1-R). Suitable hardware provides domains to isolate the kernel, and supports the simple and efficient implementation of the security kernel. For details of this approach, see Section III.

Jobstream Separator

The use of an automated approach to separating multiple levels of classified information is envisioned in a jobstream separator (JSS). The security solution is the same as for periods processing in that secure computer operations at different levels are performed in isolated time periods dedicated to a single level of classified data and user clearances. However, the major improvement over the previously described periods processing approach is in reducing the change time through automation. The color change process is controlled by attaching a minicomputer which places management and execution of change-over procedures within the internal operating environment. The application of an internal (automated) control mechanism increases

both the reliability and speed of color-changing. Response time is improved and the monitoring of uniform security levels takes place under control of a certifiably secure reference monitor in the minicomputer. The basic configuration of a computer installation that introduced a minicomputer to perform jobstream separation is shown in Figure 2.

Cost Factors

The JSS cost picture is similar to periods processing with a significantly lower change time, balanced by a small loss in machine capacity devoted to the internal security control mechanisms. In addition, all internal approaches have a development cost factor to be appropriated among future users of the kernel design implementations. The development costs for JSS, are estimated at $2-3 million, and the equipment cost at $20,000 to $30,000 per system. Finally, the lost capacity can be estimated on a basis of a 1-5 minute change period with 4 or more changes in a day.

VMM

The virtual machine monitor (VMM) approach is a method of providing functionally separate (virtual) machines for each security level while in fact affording real hardware sharing. From a security point of view, the separation of users and data is controlled internally by a reference monitor with the certifiable control logic of a kernel design. The virtual machine approach provides the same degree of multilevel service as that described under dedicated systems. The VMM approach restricts inter-level access through complete isolation of security levels; it differs from dedicated systems in that all levels are simultaneously resident in the system and security controls are logical (internal) rather than physical (external).

Cost Factors

The cost of running a secure VMM again includes both the development cost for implementing the new technology and modifying existing hardware plus the loss in machine capacity devoted to internal security controls. (See Figure 3.)

The development cost spread over all systems using secure VMM is estimated at $3-4 million. The per-system cost for modifying each machine for virtual operations is 5 to 10% of the base hardware cost but under favorable circumstances this cost could be negligible. Added to these is an estimated 5 to 15% lost capacity for system operations devoted to security controls.

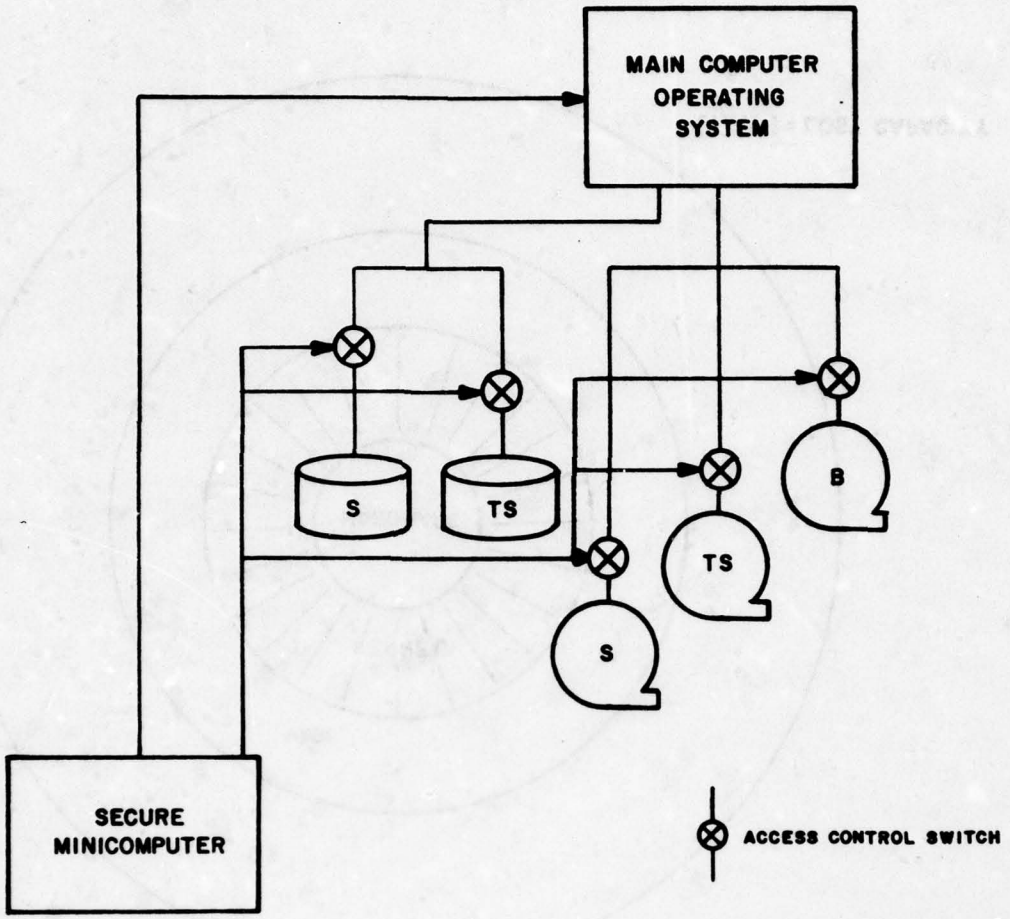


Figure 2. JOBSTREAM SEPARATOR CONFIGURATION

IA-52,283

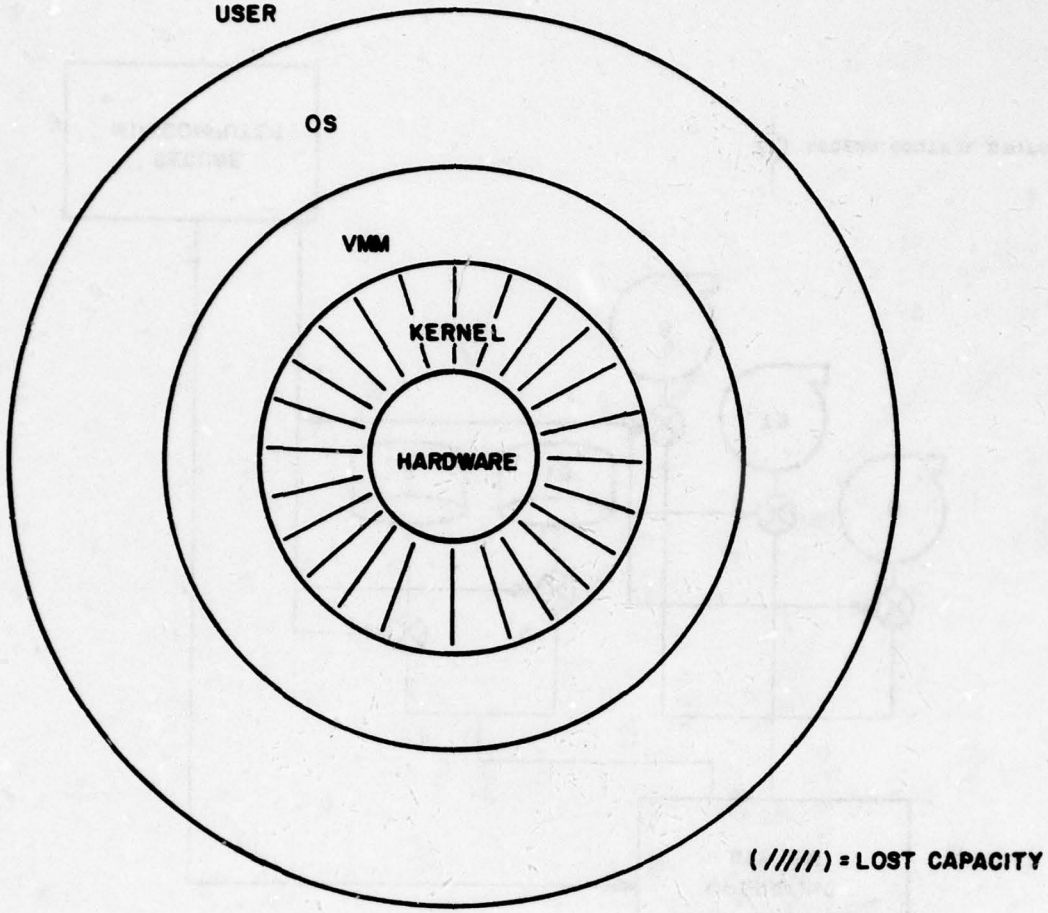


Figure 3. TYPICAL VMM ORGANIZATION

Multilevel Certifiable Secure System

The final internal approach and only real multilevel solution has as its objectives secure sharing among multiple levels of data. The multilevel approach bases its access control and data/user isolation on the functions performed by a security kernel around which a full scale multiprogramming system is designed. One prototype of the multilevel secure computer system model has currently been implemented on a minicomputer, and development stages of a large general-purpose system are in progress. The multilevel kernel approach provides the only mechanism for satisfying both hardware resource sharing and inter-level data sharing requirements.

Cost Factors

Through multilevel processing on a single machine, operational costs are reduced to a minimum; there are, however, development costs to be considered for all approaches using the new internal operating system control technology. In addition, there are per-system hardware modification costs for each installation. Both VMM and a secure multilevel approach apply the same costing algorithms to determine security overhead expense.

The development cost for a multilevel secure system is not well established, but can run from less than $1 million for a secure minicomputer operator system to several million for a large scale machine.

COMBINED APPROACHES

In defining the overall requirements of the user environment, it is often appropriate to become flexible in combining some of the approaches described above. A number of existing Air Force installations use some combination of dedicated systems with one other mode of protection. For example, a dedicated system might be used for isolating either unclassified or TOP SECRET environments while a controlled environment is used for limited multilevel processing of the other two classification levels. A job stream separator might also be applied to automated color change between SECRET and TOP SECRET while secure system is dedicated to unclassified processing. A third example is a combination systems high within periods processing. Here the isolation of data classifications according to time periods would include the processing of more than one level at the highest level for that period.

Cost Factors

The security related cost for any combination approach may be derived by appropriately combining the cost elements identified under each approach applied.

NETWORKS

One further application of the security kernel for multilevel operations is in the area of securing networks. By building secure communications processors as front-ends to host computers, it becomes feasible to provide certifiable protection for networks. Each computer on the network may be a multilevel computer or operate system-high at any one level.

Cost Factors

The cost of securing a computer network again entails a sharing of the development cost among installations using secure front-end processors in addition to any special equipment costs required for hardware modifications. Each front-end processor also has a portion of its operating capacity devoted to security controls and hence a lost capacity related to security features. Development cost is estimated at between $2 and $3 million, special equipment cost at $20,000 to $30,000 per network node and security overhead at 5 to 10%.

SECTION V

MULTILEVEL SECURITY REQUIREMENT ANALYSIS

INTRODUCTION

This section will present a concrete methodology for analyzing multilevel computer security requirements based on current or projected operational information flows. This methodology can ascertain real user requirements for multilevel operation and can also be used to identify particular problem areas where multilevel requirements can be "reduced" in a rather mechanical way.

An entire section of the report is being devoted to this particular requirement because it is felt that it is the least understood and most difficult to determine of all the requirements presented in the previous section. The technology for coping with this requirement is relatively new and many of the people to whom this report is addressed may not understand why, in some cases, it is necessary.

Origin

All ADP users do not work in a vacuum and must of necessity interact with or employ the data and programs of other users. Operating systems, the controlling programs on a computer under which users run, have in the past been primarily directed towards encouraging the sharing of information between users. They have not, until very recently, successfully addressed the problem of keeping information on a computer from users who wish to access it. In fact, no current commercial system today can protect resident information from being accessed by a sufficiently knowledgeable user who has physical access to the computer. Consequently, in military systems that contain classified data, it has been common practice to deny users physical access unless they were cleared to the highest level of information on the system. The impact of this method, however, is becoming severe since sophisticated management and programming tools cannot be used for security reasons. The complexity of data sharing necessary in modern computer systems to effectively support military users in their task of managing and maintaining their forces and equipment is often great. Large data bases with many levels of classified data must be resident on these systems and they must, in many cases, be accessible to users of various clearance levels, implying a need for a multilevel secure system.

The requirement for multilevel computer operations is, in general, driven by operational requirements that require users to access a system in which information classified higher than their clearance level is resident. If such users must have access to the system (by having the capability to submit jobs to the system either via a terminal, card deck or some other means), then the internal access controls of the system must be capable of protecting the classified data from compromise. The purpose of the methodology presented here, then, is to determine if such operational requirements do exist.

The distinction between multilevel integrated processing requirements (or equivalently, data sharing) requirements driving multilevel operation and the security requirements associated with such operation should be noted; it is an important one. The former, the subject of this section, represents a determination of the need for multilevel operation based on mission requirements. The latter, on the other hand, deals with technical and procedural ramifications of a decision to operate in a multilevel mode and is treated in Section III.

The decision to operate in a multilevel mode need not be based on multilevel data sharing requirements; economic considerations, for instance, may suggest multilevel operation. Of the available approaches listed in Section IV, several, in fact, permit a form of multilevel operation that is functionally equivalent to the use of separate machines for each security level. As such, these approaches cannot address the integrated processing requirements presently under discussion.

Examples

As a simple example of an operational situation in which a multilevel requirement is present, consider the two-user two-data file system illustrated in Figure 4A. Let us make the assumption that:

1. the Top Secret user is involved in activities that require essentially simultaneous access to current copies of both files (e.g., he is required to update one from the other), and

2. the Secret user needs to access a current copy of the Secret file.

In this case, then, a multilevel requirement does exist and the degree of potential compromise (or the degree of data sharing)

will be defined as S-TS; that is, a user cleared to the Secret level must, of necessity, have access to a system that contains Top Secret information.[9]

Figures 4B and 4C, on the other hand, present operational situations in which there is no multilevel data sharing requirement. In Figure 4B, the secure operation on a computer system could be accomplished simply by running at a Secret level. The highest level of information that could be accessed by a user is Secret, a level for which both users are cleared. It would, of course, be incumbent on the TS user to ensure that information be introduced into the system would be classified Secret or lower but this restriction is not unusual and is in fact done on most present one level systems.

The situation pictured in Figure 4C has no multilevel sharing requirement and would be amenable to isolation approaches to achieve security. If each of the users and their files were isolated, either by putting them on different systems or by employing periods processing to isolate them on the same system, then security could be maintained.

Figure 4D illustrates a more complex situation where there appears to be a multilevel requirement between C and TS levels. Under certain circumstances, however, this example might be adapted to two environments, one with S-TS multilevel sharing and another with no sharing, running at a confidential level. In particular, if the Secret user was not required to access both the Secret and Confidential files at the same time then these files could be put in separate environments each of which the Secret user could access. The primary advantage of splitting the ADP environment in two pieces is primarily to reduce the potential degree of compromise and will be discussed more fully in the next section. The important point to be drawn from this example, however, is that the collection of files a user accesses will not be as critical to multilevel sharing requirements as the operational interaction of files.

In many situations, operational requirements may not dictate that all users have up-to-the-minute versions of data files. For long range planning purposes, for example, some users may require only a weekly updated version of certain data. Because these users must access other sensitive, highly classified data, they may be introducing unnecessarily broad multilevel data sharing requirements

---

[9] This, of course, is not to say that he must or could have access to the TS file.

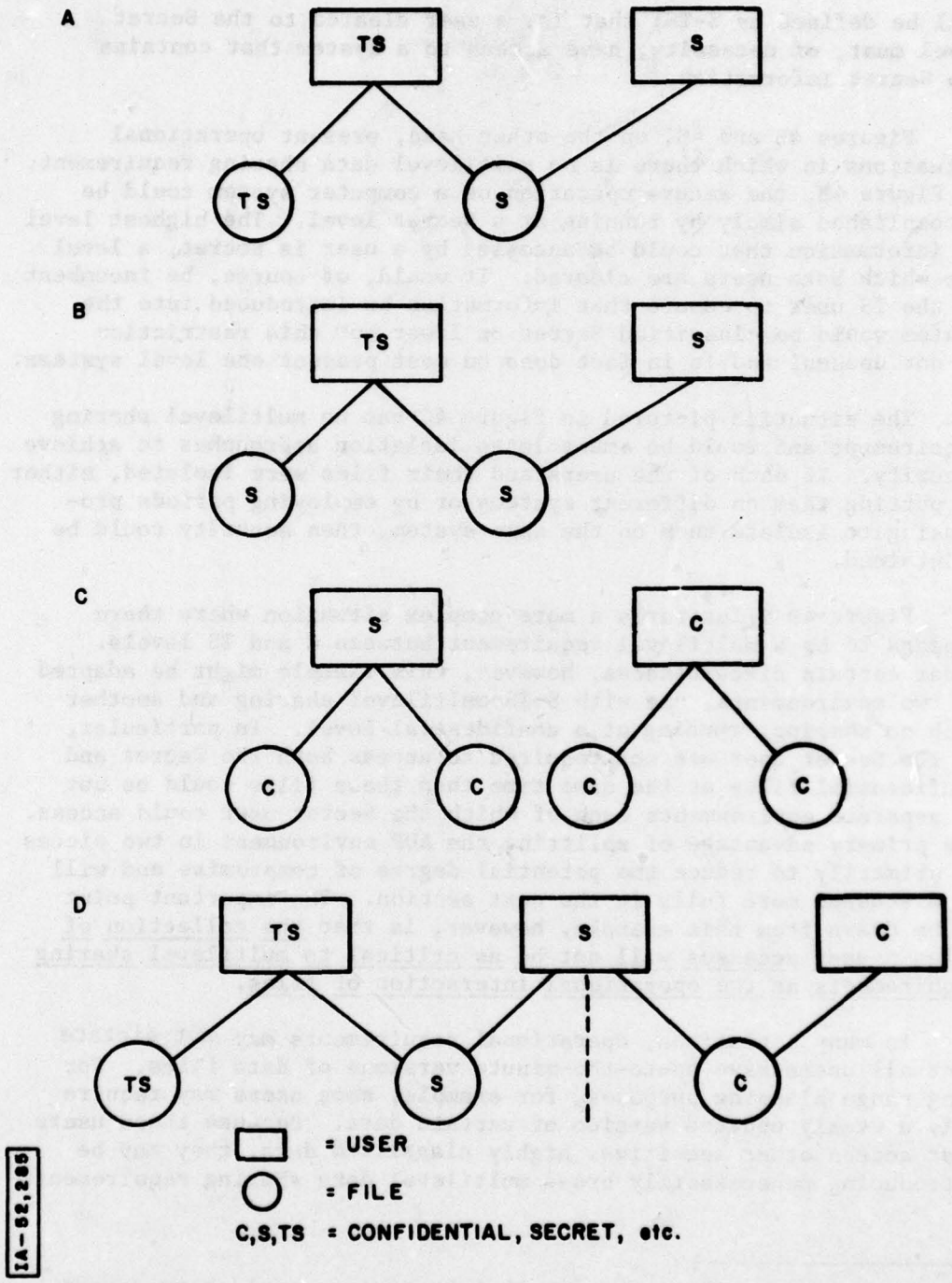


Figure 4. MULTILEVEL REQUIREMENTS EXAMPLES

into the day-to-day operational environment. This "data concurrency" problem and its solution can best be illustrated with a concrete example.

Figure 5A presents a two user, two file system. If one assumes that the Secret user requires access to information in the Confidential file only on a monthly basis, then this configuration can be taken as an example of the data currency problem. That is, while the Secret user may require access to the Confidential file only once per month, he induces an apparent C-S multilevel data sharing requirement into all operations.

As Figure 5B indicates, the most direct approach to dealing with data currency problems is simply to periodically duplicate critical files to reduce the amount of "dynamic" multilevel sharing among users. While users may be logically sharing information, the files that they physically access in the course of operations can be distinct. Therefore, in an operational sense, multilevel sharing requirements can be reduced by means of file duplication. The obvious tradeoff is the effort and expense of duplicating files, thus it will be generally impractical to eliminate all sharing requirements among users. The general motivation for partitioning groups of users and files and the potential advantages of splitting up groups of users in the above manner will be treated in the following paragraphs.

User/File Group Partitioning

Basic approaches allowing the secure processing of information at several classification levels have been discussed in Section II. Of these approaches, only the use of controlled environments, multilevel secure operating systems, and to a much less extent, the system high approach, permit multilevel information sharing. All other approaches rely on the isolation of users and files at different classification levels. Consequently, the impact of multilevel sharing requirements on system designers is quite severe--in many cases unnecessarily so.

As some of the examples of the previous section have indicated, the division of the ADP environment into distinct groups of files and users can:

1. Reduce or eliminate multilevel sharing requirements within individual groups (that is, while the entire environment might require multilevel operation in an Unclassified to Top Secret range, a particular group of users might require only Secret to Top Secret operation of dedicated Top Secret operation).

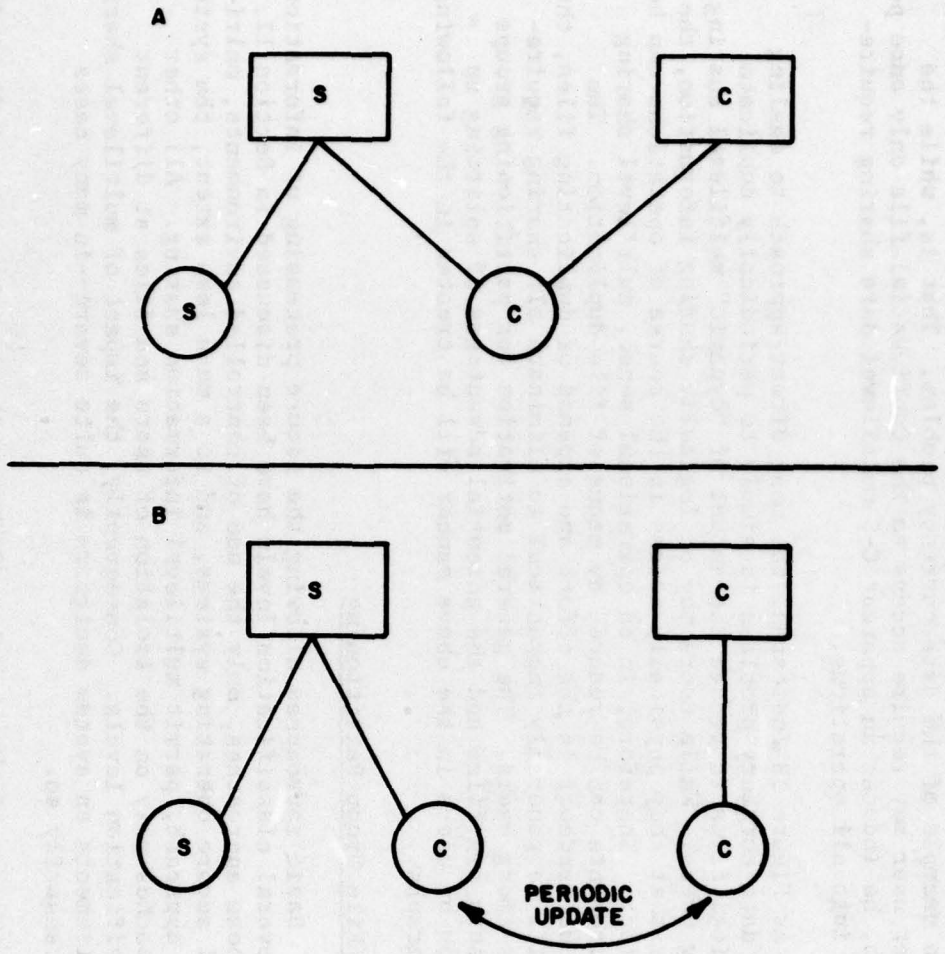


Figure 5. DATA CURRENCY

2. Enable application of more easily implemented approaches to (externally or internally) isolate these groups.

Any division, of course, must insure that no files in two distinct groups are not required to directly interact with one another (unless one may be copied and periodically updated). A user, however, may participate in more than one group, assuming he is sufficiently cleared, simply by moving to a new environment.

Before any firm commitments are made towards one approach or a particular combination of approaches, the structure of the file and user groups and the overall division of the ADP environment should be examined to determine whether relatively easy and inexpensive adjustments can be made to reduce multilevel sharing requirements.

REFERENCES

Air Force Regulations, "Information Security Program", AFR-205-1, February 1973, AFSC Supplement July 1973, ESD Supplement, January 1974.

Department of Defense, "Security Procedures for Automatic Data Processing Systems", DoD Directive 300-8, September 1974.

American Federation of Information Processing Societies, "AFIPS 1974 System Review Manual on Security", AFIPS Press, July 1974.

Alexander, T., "Waiting for the Great Computer Rip-off", Fortune, Vol XC, No. 1, July 1974, pp. 142-150.

Ames, S.R., "File Attributes and their Relationship to Computer Security", Masters' Thesis, Case Western Reserve University, ESD-TR-74-191, June 1974 (AD A002159).

Ames, S. R., "The Design of a Security Kernel", The MITRE Corporation, M75-212, April 1975.

Anderson, J. P., "Computer Security Technology Planning Study", James P. Anderson and Co., ESD-TR-73-51, Volume I & II, October 1972 (Ad 758206 and AD 772806).

Anderson, J. P., "Multics Evaluation", James P. Anderson and Co., ESD-TR-73-276, October 1973 (AD 777593).

Bell, D. E. and Burke, E. L. "A Software Validation Technique for Certification, The Methodology", ESD-TR-75-54, Electronic Systems Division, AFSC, Hanscom AFB, MA, April 1975 (AD A009849).

Bell, D. E. and LaPadula, L. J., "Secure Computer Systems: Mathematical Foundations and Model", The MITRE Corporation, M74-244, October 1974.

Bell, D. E., and LaPadula, L. J., "Secure Computer System: Unified Exposition and Multies Interpretation", ESD-TR-75-306, Electronic Systems Division, AFSC, Hanscom AFB, MA, March 1976 (AD A023588).

Branstad, D. "Privacy and Protection in Operating Systems", Computer, Volume 6, Number 1, January, 1973, pp. 43-47.

Burke, E. L., "Concept of Operation for Handling I/O Input/Output in a Secure Computer at the Air Force Data Services Center AFDSC", ESD-TR-74-113, Electronic Systems Division, AFSC, Hanscom AFB, MA, April 1974 (AD 780529).

Burke, E. L., Synthesis of a Software Security System", Proceedings of ACM 74 Conference, MTP-54, November 1974.

Burke, E. L. et al, "Emulating a Honeywell 6180 Computer System", The MITRE Corporation, RADC-TR-74-137, June 1974 (AD 787218).

Burke, E. L. et al, "Secure Minicomputer Systems Architecture", Proceedings of the IEEE Compcon 74 Fall, Washington, D.C., September 1974.

Clark, B. L. The Design of a System Programming Language, M.Sc. Thesis, University of Toronto, Toronto, Canada, November, 1971.

Dijkstra, E. W., "The Structure of the 'THE' Multiprogramming System", Communications of the ACM, Volume 11, Number 5, May, 1968, pp. 341-346.

Department of Defense, "Security Requirements for Automatic Data Processing (ADP) Systems", DoD 5200.28, December 1972.

Department of Defense, "Security Manual: Technique and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing ADP Systems", DoD 5200.28M, January 1973.

Goldberg, R. P., "Architectural Principles for Virtual Computer Systems", Ph.D. Thesis, Harvard University, ESD-TR-73-105, February 1973 (AD 772809).

Graham "Protection in an Information Processing Utility "Communications of the ACM", May 1968, pp. 365-369.

Honeywell Information Systems, Cambridge, MA, Design For Multics Security Enhancements, ESD-TR-74-176, Electronic Systems Division, AFSC, Hanscom AFB, MA, December 1977, (A 030801).

Janson, P. A., "Removing the Dynamic Linker from the Security Kernel of a Computing Utility", Massachusetts Institute of Technology, Project MAC TR-132, June 1974.

Karger, P. A. and Schell, R. R., "Multics Security Evaluation: Vulnerability Analysis", ESD-TR-74-193, Volume II, Electronic Systems Division, Hanscom AFB, MA, June 1974 (AD A001120).

Lampson, B. W. "A Note on the Confinement Problem", Communications of the ACM, Volume 16, Number 10, October, 1973, pp. 613-615.

Lipner, S. B., "A Minicomputer Security Control System", The MITRE Corporation, MTP-151, February 1974.

Lipner, S. B., "Computer Security Research and Development Requirements", The MITRE Corporation, MTP-142, February 1973 (out of print).

Lipner, S. B., "Multics Security Evaluation: Results and Recommendations", The MITRE Corporation, ESD-TR-74-193, Volume I (in progress), June 1974.

Lipner, S. B., "Panel Overview (A panel session - Security Kernels)", AFIPS Conference Proceedings, Vol. 43 (1974 NCC), p. 973-974).

Liskov, B. H. "The Design of the Venus Operating System", Communications of the ACM. Volume 15, Number 3, March, 1972, pp. 144-149.

Mogilensky, J., "A General Security Marking Policy for Classified Computer Input/Output Material", ESD-TR-75-89, Electronic Systems Division, AFSC, Hanscom AFB, MA, September 1975 (AD A016467).

Multics Programmers' Manual, AG91, AG92, AG93 and AK92, Honeywell Information Systems Inc., 1975.

Anderson, J. P., "Systems Architecture for Security and Protection", NBS Special Publication 404, September 1974, pp. 45-50.

Lipner, S. B., "Security Considerations in Information Design", NBS Publication 404, September 1974, pp. 55-59.

Organick, E. I., The Multics System: An Examination of Its Structure, MIT Press, Cambridge, MA, 1972.

Parnas, D. L., "A Technique for Software Module Specification with Examples", Communications of the ACM, Volume 15, Number 5, May 1972, pp. 330-336

Popek, G. J., "Access Control Models", Ph.D. Thesis, Harvard University, ESD-TR-73-106, February 1973 (AD 761807).

Price, W. R., Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, June, 1973.

Rhode, R. D., "Secure Multilevel Virtual Computer Systems", ESD-TR-74-370, Hanscom AFB, MA, February 1975 (AD A007059).

Robinson, L., et al, "On Attaining Reliable Software for a Secure Operating System", 1975 International Conference on Reliable Software, Los Angeles, California, April, 1975, pp. 21-23.

Saltzer, J. H., "Protection and the Control of Information in Multics", Communications of the ACM, Volume 17, Number 7, July, 1974, pp. 388-402.

Saltzer, J. H., Traffic Control In a Multiplexed Computer System, MAC-TR-30 (Ph.D. Thesis), MIT Project MAC, Cambridge, MA, July 1966.

Schell, R. R., "Effectiveness - The Reason for a Security Kernel (A panel session - Security Kernels)", AFIPS Conference Proceedings, Vol. 43, (1974 NCC), p. 975-976.

Schacht, J. M., "Jobstream Separator System Design", ESD-TR-75-86, Electronic Systems Division, AFSC, Hanscom AFB, MA, September 1975 (AD A016403).

Schiller, W. L., "Design of a Security Kernel for the PDP-11/45", ESD-TR-73-294, Hanscom AFB, MA, December 1973 (AD 772808).

Schiller, W. L., "The Design and Specification of a Security Kernel for the PDP-11/45", ESD-TR-75-69, Electronic Systems Division, AFSC, Hanscom AFB, MA, May 1975 (AD A011712).

Smith, L. A., "Architectures for Secure Computing Systems", ESD-TR-75-51, Electronic Systems Division, AFSC, Hanscom AFB, MA, April 1975 (AD A009221).

Stork, D. F., "Downgrading in a Secure Multilevel Computer System: The Formulary Concept", ESD-TR-75-62, Electronic Systems Division, AFSC, Hanscom AFB, MA, May 1975 (AD A011696).

Walter, K. G. et al, "Modeling the Security Interface", Department of Computing and Information Sciences, Case-Western Reserve University, Cleveland, Ohio, August 1974.

Walter, K. G. et al, "Primitive Models for Computer Security", Department of Computing and Information Sciences, Case-Western Reserve University, Cleveland, Ohio, ESD-TR-74-117, January 1974 (AD 778467).

Walter, K. C., et al, "Models for Secure Computer Systems", Report No. 1137, Department of Computing and Information Sciences, Case-Western Reserve University, Cleveland, Ohio, July 1973.

Weissman, C. "Security Controls in the ADEPT-50 Time-Sharing System", Proceedings AFIPS 1969 FJCC, AFIPS Press, Montvale, New Jersey, 1969. pp. 119-133.

White, J. C. C., "Design of a Secure File Management System", ESD-TR-75-57, Electronic Systems Division, AFSC, Hanscom AFB, MA, April 1975 (AD A010590).

Wulf, W. et al, "HYDRA: The Kernel of a Multiprocessor Operating System", Communications of the ACM, Volume 17, Number 6, June 2974, pp. 337-345.